

Projeto de Fundamentos de IoT Vagas no Estacionamento

Reynaldo Caceres Villena

IME-USP

Junho - 2018

1 Introdução

2 Estrutura de Projeto

- Aparelhos
 - Raspberry Pi 3 Model B
 - Câmeras
- Serviços
 - Computer Visual API- Azure
 - Pythonanywhere
 - Linguagens de Programação
- Arquitetura do Projeto
- Experimentos
 - Aplicação de Machine Learning

3 Conclusões

Introdução

Vagas no Estacionamento



1 Introdução

2 Estrutura de Projeto

- Aparelhos
 - Raspberry Pi 3 Model B
 - Câmeras
- Serviços
 - Computer Visual API- Azure
 - Pythonanywhere
 - Linguagens de Programação
- Arquitetura do Projeto
- Experimentos
 - Aplicação de Machine Learning

3 Conclusões

Raspberry Pi 3 Model B



- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1 & Bluetooth Low Energy (BLE)
- 1GB RAM
- 10/100 Ethernet port
- 802.11n WiFi NIC
- Bluetooth 4.1 & Bluetooth Low Energy (BLE)
- HDMI port
- USB 2.0 interface x 4
- Micro SD card slot
- 40-pin GPIO interface
- Camera interface (CSI)
- Display interface (DSI)
- Upgraded power management
- supports more peripherals

Câmeras



Camera Multilaser WC040

- Vídeos de 1.3 Mpx (1280 x 1024 pixels)
- Fotos de 5 Mpx (via interpolação)
- Conexão direta (USB).



Celular LG G3

- Vídeos em 4K (3840x2160 pixels)
- Fotos de 13 Mpx (4160x3120 pixels)
- Conexão via LAN (aplicativo IP Webcam)

Serviços

Computer Vision API

Mycrosoft Azure (<https://azure.microsoft.com/en-us/services/>)

The screenshot shows the Microsoft Azure website's "AI + Machine Learning" section. On the left, there's a sidebar with categories like Compute, Networking, Storage, etc. The main content area has a heading "AI + Machine Learning" with a "Learn more >" link. Below it, several services are listed in pairs:

- Machine Learning Studio: Easily build, deploy, and manage predictive analytics solutions.
- Cognitive Services: Add smart API capabilities to enable contextual interactions.
- Azure Bot Service: Intelligent, serverless bot service that scales on demand.
- Text Analytics API: Easily evaluate sentiment and topics to understand what users want.
- Computer Vision API**: Distill actionable information from images.
- Content Moderator: Automated image, text, and video moderation.
- Emotion API: Personalize user experiences with emotion recognition.
- Face API: Detect, identify, analyze, organize, and tag faces in photos.
- Bing Speech API: Convert speech to text and back again to understand user intent.
- Web Language Model API: Use the power of predictive language models trained on web-scale data.

- Computer Vision API tem um conjunto de algoritmos que processam as imagens e devolvem certa informação.

Serviços

Computer Vision API

See it in action

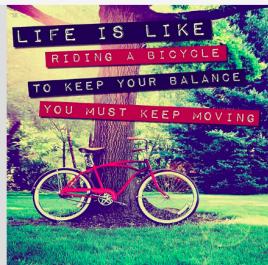


Image URL Submit

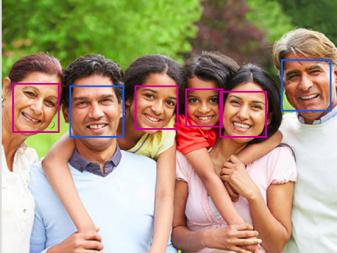
FEATURE NAME:	VALUE
Description	[{"tags": ["train", "platform", "station", "building", "indoor", "subway", "track", "walking", "waiting", "pulling", "board", "people", "man", "luggage", "standing", "holding", "large", "woman", "yellow", "suitcase"], "captions": [{"text": "people waiting at a train station", "confidence": 0.833099365}]}]
Tags	[{"name": "train", "confidence": 0.9975446}, {"name": "platform", "confidence": 0.995543063}, {"name": "station", "confidence": 0.9798007}, {"name": "indoor", "confidence": 0.927719653}, {"name": "subway", "confidence": 0.838939846}, {"name": "pulling", "confidence": 0.431715637}]
Image format	"jpeg"

Preview JSON

LIFE IS LIKE
RIDING A BICYCLE
TO KEEP YOUR BALANCE
YOU MUST KEEP MOVING



See it in action



FaceRectangles [{"age": 40, "gender": "Male", "faceRectangle": {"top": 159, "left": 119, "width": 93, "height": 931}, "age": 81, "gender": "Male", "faceRectangle": {"top": 110, "left": 490, "width": 92, "height": 921}, "age": 60, "gender": "Female", "faceRectangle": {"top": 152, "left": 86, "width": 86, "height": 861}, "age": 35, "gender": "Female", "faceRectangle": {"top": 186, "left": 386, "width": 82, "height": 821}, "age": 30, "gender": "Female", "faceRectangle": {"top": 157, "left": 234, "width": 79, "height": 791}, "age": 10, "gender": "Female", "faceRectangle": {"top": 162, "left": 322, "width": 68, "height": 681}]

Dominant color background □ White

Dominant color foreground □ White

FEATURE NAME:	VALUE
Description	[{"tags": ["car", "building", "road", "sitting", "parked", "small", "city", "street", "table", "mirror", "man", "police", "white", "parking", "riding", "driving", "bed", "people", "kitchen", "traffic", "blue"], "captions": [{"text": "a car parked on the side of a building", "confidence": 0.953054667}]}]
Tags	[{"name": "car", "confidence": 0.9977151}]
Image format	"jpeg"
Image dimensions	525 x 700
Crop art	0

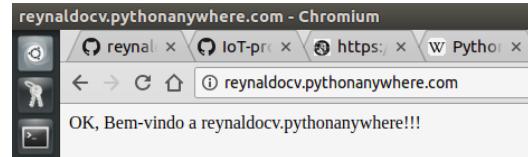
Serviços

Pythonanywhere

<https://www.pythonanywhere.com/>



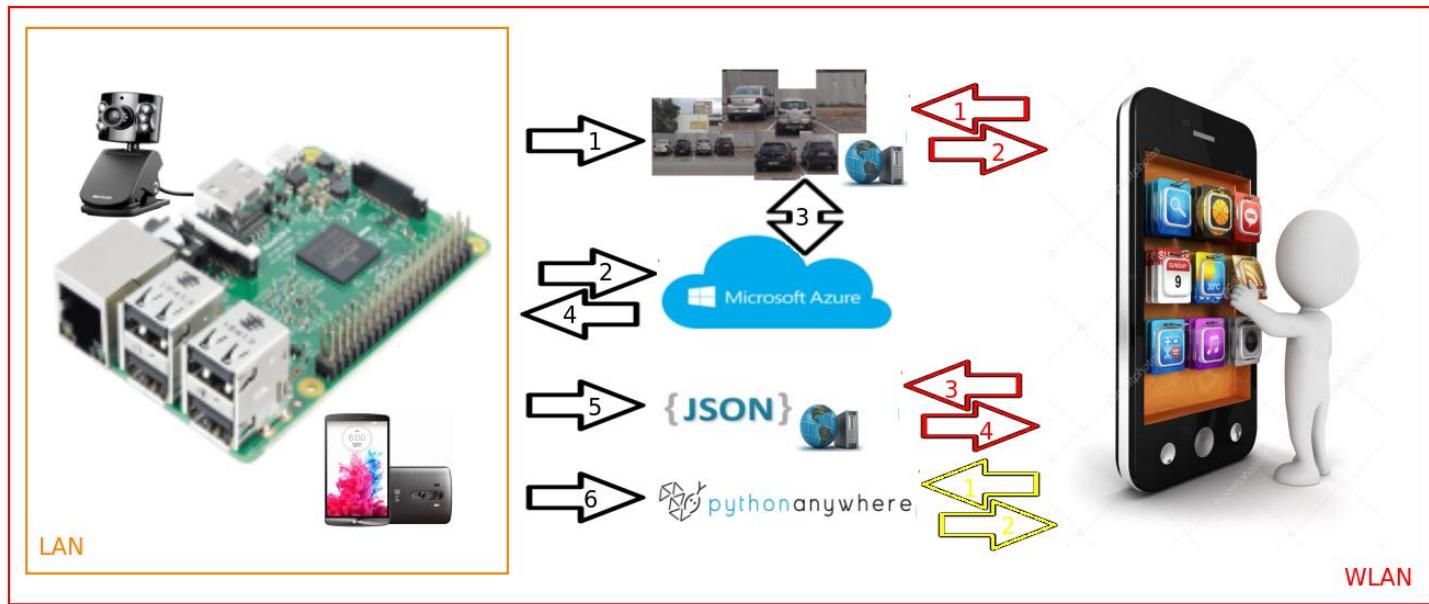
- Ambiente de desenvolvimento integrado (IDE)
- Serviço de hospedagem web



Linguagens de Programação

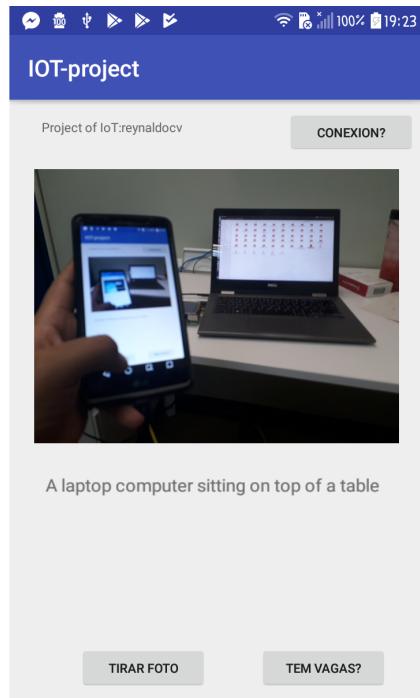
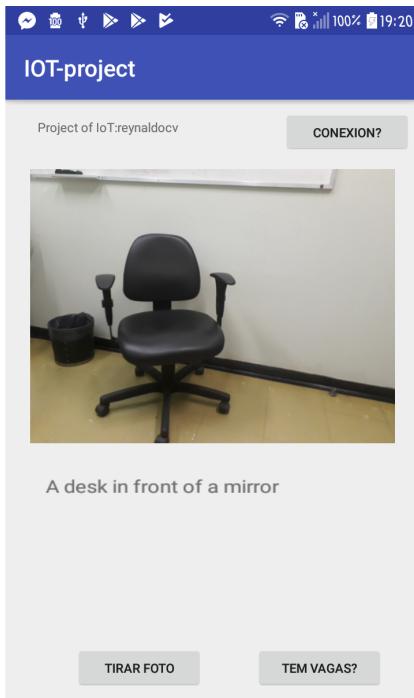
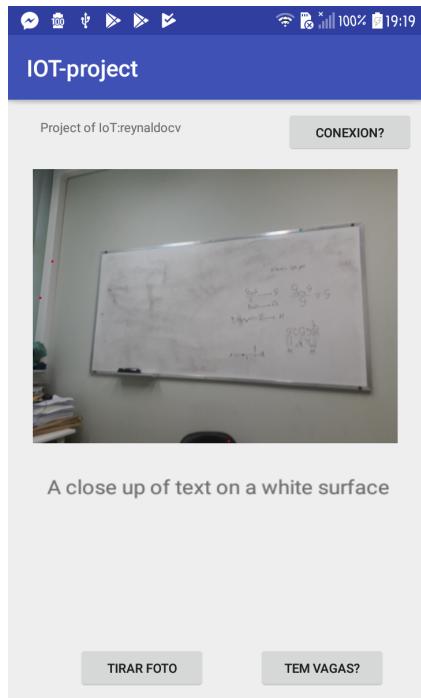


Arquitetura do Projeto



Experimentos

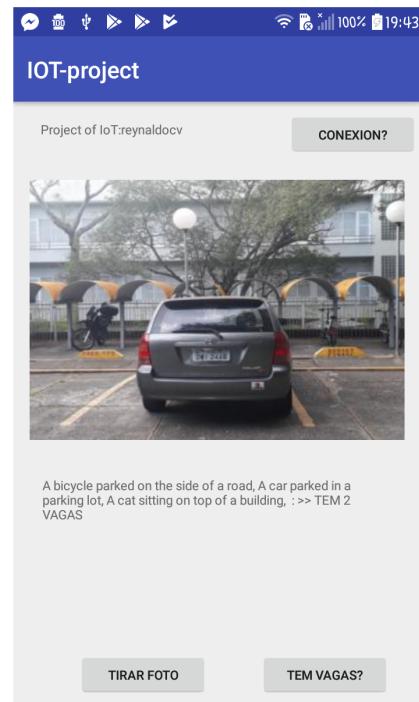
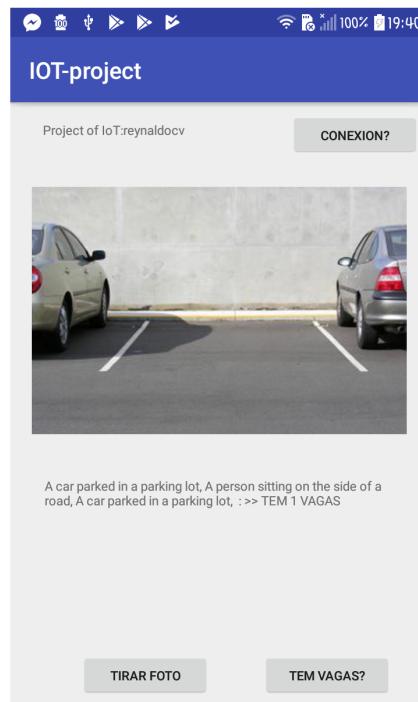
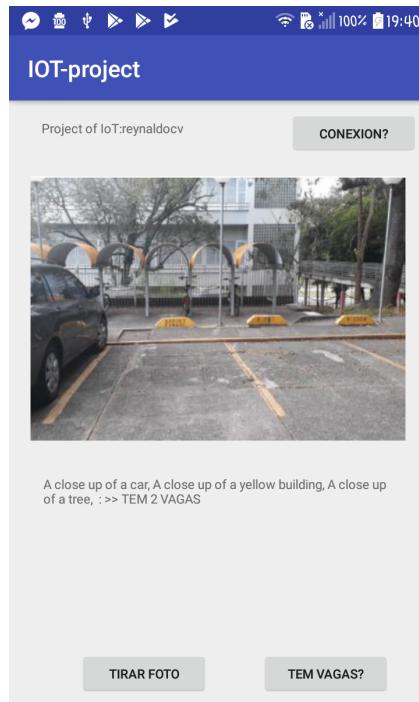
Funcionamento do API Computer Vision



- Conexão direta entre o Raspberry e o Celular (mesma LAN)

Experimentos

Funcionamento do API Computer Vision



- Conexão direta entre o Raspberry e o Celular (mesma LAN)

Experimentos

Funcionamento com carrinhos



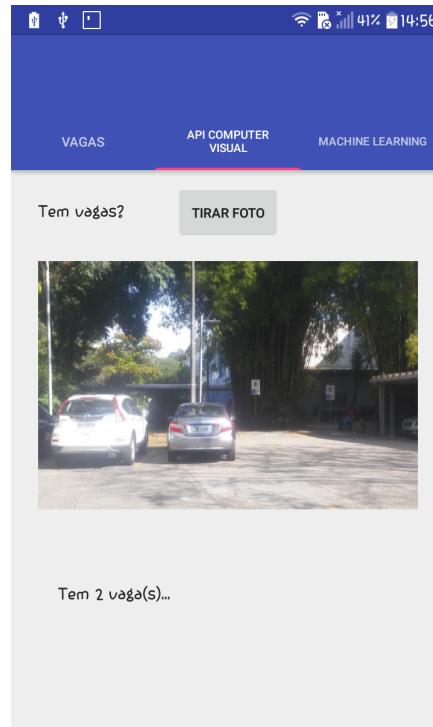
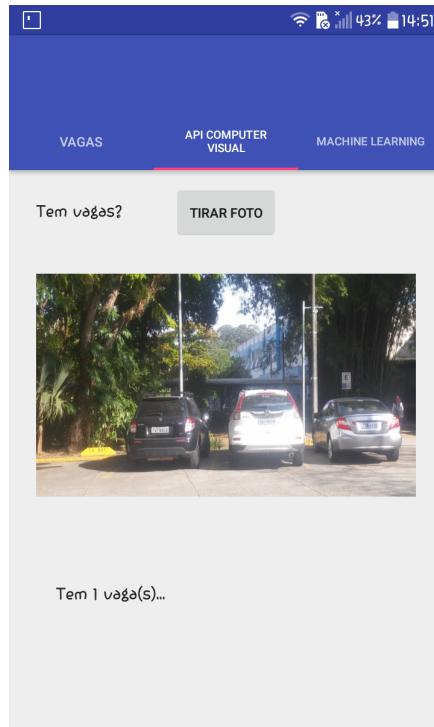
Experimentos

Funcionamento com carrinhos



Experimentos

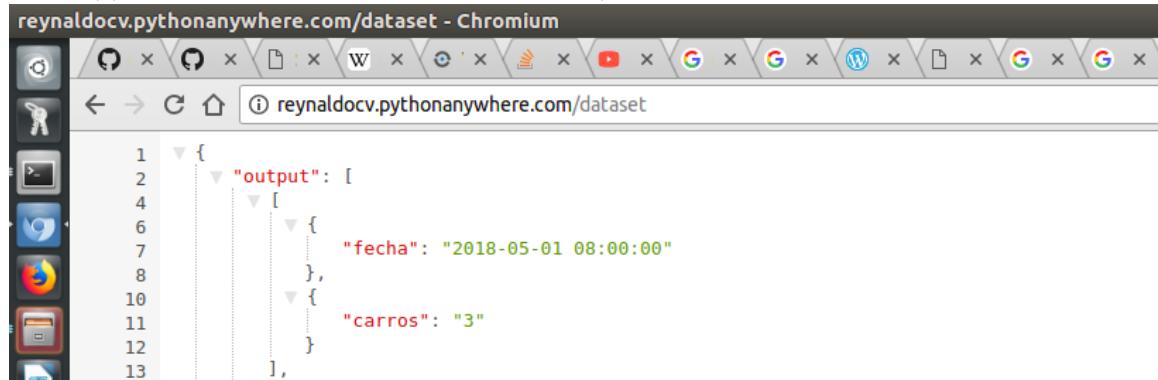
Funcionamento



Aplicação de Machine Learning

APIs criadas no pythonanywhere:

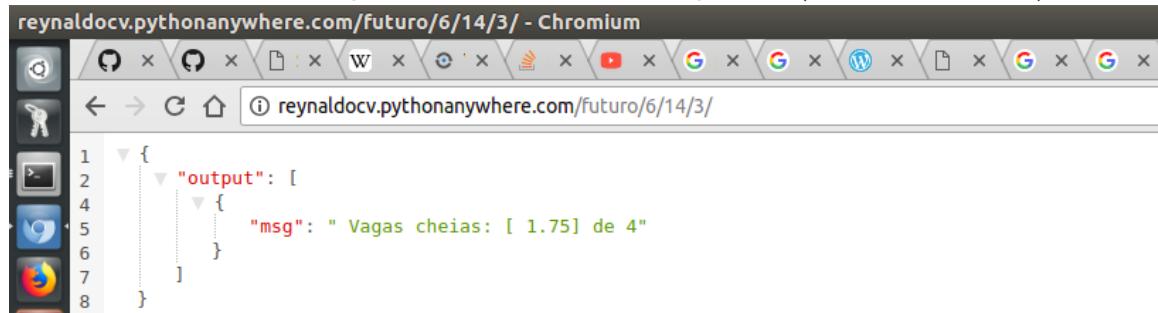
- `http://reynaldoctrl.pythonanywhere.com/add/ <text1>`
- `http://reynaldoctrl.pythonanywhere.com/dataset`



The screenshot shows a Chromium browser window with the URL `reynaldoctrl.pythonanywhere.com/dataset`. The page displays a JSON response with the following structure:

```
1  {
2   "output": [
3     [
4       {
5         "fecha": "2018-05-01 08:00:00"
6       },
7       {
8         "carros": "3"
9       }
10    ],
11  ],
12  ],
13 ]
```

- `http://reynaldoctrl.pythonanywhere.com/futuro/ <dia>/<hora>/<minutos>`
Biblioteca sklearn, e o algoritmo Decision-Tree-Regressor (Acurácia de 67%)



The screenshot shows a Chromium browser window with the URL `reynaldoctrl.pythonanywhere.com/futuro/6/14/3/`. The page displays a JSON response with the following structure:

```
1  {
2   "output": [
3     {
4       "msg": " Vagas cheias: [ 1.75] de 4"
5     }
6   ],
7 },
8 ]
```

Experimentos(Junho- 2018)

Consulta de vagas no futuro:

VAGAS API COMPUTER VISUAL MACHINE LEARNING

vag&as no dia: Sabado
às: 12 hrs 45 min

PREDECIR

Vagas cheias: [1.75] de 4

VAGAS API COMPUTER VISUAL MACHINE LEARNING

vag&as no dia: Quarta feira
às: 13 hrs 50 min

PREDECIR

Vagas cheias: [2.25] de 4

VAGAS API COMPUTER VISUAL MACHINE LEARNING

vag&as no dia: Sexta feira
às: 14 hrs 25 min

PREDECIR

Vagas cheias: [4.] de 4

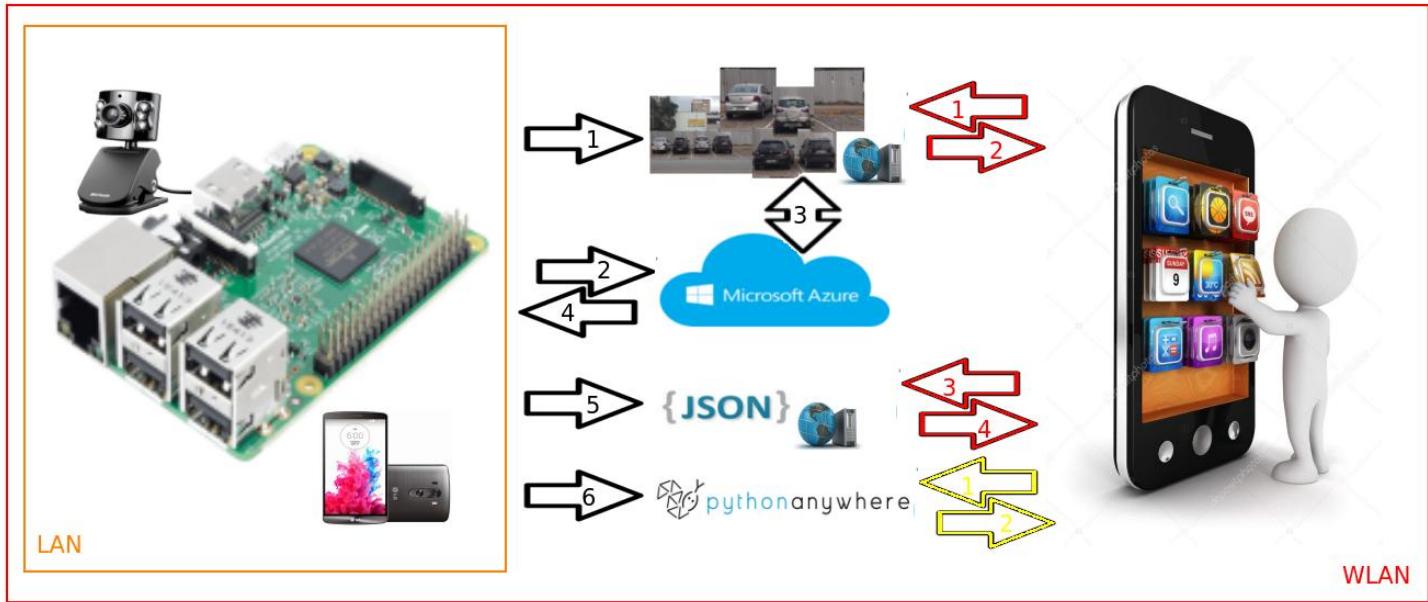
1 Introdução

2 Estrutura de Projeto

- Aparelhos
 - Raspberry Pi 3 Model B
 - Câmeras
- Serviços
 - Computer Visual API- Azure
 - Pythonanywhere
 - Linguagens de Programação
- Arquitetura do Projeto
- Experimentos
 - Aplicação de Machine Learning

3 Conclusões

Conclusões



- No procedimento, existe uma inconsistência da informação durante uns 30 segundos até 2 min.
- A computer Vision API - Azure é uma boa ferramenta, mas precisa imagens de boa resolução.

Obrigado!!!



Toda informação e código deste projeto pode ser acessado em:

- <https://github.com/reynaldocv/IoT-project>